IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appl. No.	:	**09/801,080**
Conf. No.	:	**5082**
Applicant(s)	:	**NATALINO GIORGIO BUSA ET AL.**
Filed	:	**7 MARCH 2001**
TC/A.U.	:	**2183**
Examiner	:	**DANIEL H. PAN**
Atty. Docket	:	**NL-000133**
Title	:	**DATA PROCESSING DEVICE, METHOD OF OPERATING A DATA PROCESSING DEVICE AND METHOD FOR COMPILING A PROGRAM**

REPLY BRIEF

Mail Stop **Appeal Brief - Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Examiner's Answer mailed on 22 May 2008, Appellants herewith respectfully present their Reply Brief as follows:

ARGUMENT/REMARKS

Appellants respectfully assert that the Examiner's Answer ("Answer") fails to cure the factual and legal deficiencies set forth in the Final Office Action, as demonstrated in Appellants' Appeal Brief. Appellants maintain that claims 1-5 are patentable and allowable over the cited art of record.

O'Conner does not Anticipate the Claimed inventions

In the Response to Argument section of the Answer, the Examiner responds to Appellants' arguments regarding O'Conner by noting that "nowhere does appellant's claim recite 'concurrently execution'" (see Answer, p. 8, section (10)). However, in the Appeal Brief, Appellants distinguish the subject matter of claims 1 and 4, for example, from O'Conner by explaining that:

> **In O'Connor, one execution unit waits for results from another execution unit in order for the first execution unit to begin execution. The execution units do not operate concurrently to achieve execution of an instruction.**

In view of the above, the Examiner's assertion that "nowhere does appellant's claim recite 'concurrently execution'" is not well founded, and simply fails to address Appellants arguments.

Indeed, to be clear, Appellants make no specific assertion in the Appeal Brief regarding "concurrently execution", as noted in the Answer. Rather, as shown above, Appellants maintain that **the execution units (in O'Conner) do not operate concurrently** to achieve execution of an instruction, which is one clear distinction between O'Conner and the inventions of claims 1 and 4, for example.

More specifically, claim 1 recites, in part, a data processing device " . . .being programmed for executing an instruction by the first functional unit (including a slave controller), execution of said instruction involving input/output operations by the first functional unit, wherein said execution involves at least one of: output data of the first functional unit being processed by the second functional unit during execution of said instruction, and the input data to the first functional unit being generated by the second functional unit during execution of said instruction." (Emphasis added).

In other words, in view of the above language of claim 1, the subject matter of claim 1 can be fairly characterized as being directed to a data processing device in which first and second functional units operate concurrently during execution of the instruction by the first functional unit, as the second functional unit (i) generates and inputs data to the first functional unit during execution of the instruction by the first functional unit or (ii) processes output data received from the first functional unit, during execution of the instruction by the first functional unit.

The Examiner maintains on page 8 of the Answer that:

> Nevertheless, even if one execution unit had to wait for another execution unit, it did not necessarily mean that it was not a concurrent processing. A concurrent processing could be two execution units run independently and dependently at respective points in time, and produced results simultaneously.

Again, Appellants respectfully assert that the Examiner's argument in this regard is misplaced and fails to understand the Appellants contention that **the execution units (in O'Conner) do not operate concurrently** to achieve execution of an instruction by the first functional unit. If one execution is "waiting" to receive results or data from another execution unit, the execution unit in a "wait" state cannot

reasonably be considered to be "operating".

Furthermore, the Examiner's characterization of O'Conner as applied to the claimed inventions is clearly erroneous for various other reasons, as explained hereafter.

For instance, claim 1 is directed to a data processing device, which comprises, e.g., a master controller, a first functional unit including a slave controller, and a second functional unit. The Examiner maintains (on pg. 5-6 of the Answer) that O'Conner discloses in FIG. 1 a "master controller" (i.e. a MUX) and "a first function unit including a slave controller" (i.e., Execution Unit (10) controlled by MUX) and a "second function unit" (i.e., Execution unit (20)). However, the Examiner's characterization of O'Conner as applied to the features of claim 1 in this regard is factually erroneous.

For instance, there is no reasonable basis to find that any MUX in FIG. 1 of O'Conner is a "master controller" within the scope and meaning of claim 1. In the context of the claimed inventions, a "master controller" controls overall operations (functional units) of the data processing device. The MUXs in FIG. 1 of O'Conner are nothing more than electronic switches that selectively apply data at one of the MUX inputs to respective Execution units. One of ordinary skill in the art would not remotely consider any MUX as a "master controller". In fact, given that each MUX in FIG. 1 only operates to input data to one of the Execution units, no MUX in FIG. 1 can even be construed as a "master" MUX that inputs data to all functional units, much less a "master controller" of functional units, within the scope of the claimed

inventions.

Moreover, there is no basis for the Examiner's finding that O'Conner discloses "a first functional unit including a slave controller". Indeed, there is no teaching, whatsoever, in O'Conner that the Execution unit (10) ("first functional unit") includes a "slave controller". The Examiner seemingly argues that such claim feature is disclosed in FIG. 1 where the execution unit (10) is controlled by a MUX (see bottom of page 5 of Answer). However, this finding is not well founded.

To begin, no MUX in FIG. 1 of O'Conner is "included" in a functional unit (i.e., no disclosure of a first functional unit including a slave controller"). Indeed, the MUXs in FIG. 1 are clearly separate, and not included, in any one of the execution units 10, 20. Moreover, the Examiner's characterization of a MUX being a "slave controller" squarely contradicts the Examiner initial finding the MUX is a "master controller." Clearly, without doubt, the Examiner's characterization of the MUX being a "master controller" and "slave controller" renders the Examiner's anticipation analysis factually flawed on its face.

Moreover, O'Conner does not disclose (in Col. 1, lines 16-17 and lines 28-32) a data processing device in which first and second functional units operate concurrently during execution of the instruction by the first functional unit, as the second functional unit (i) generates and inputs data to the first functional unit during execution of the instruction by the first functional unit or (ii) processes output data received from the first functional unit, during execution of the instruction by the first functional unit, as substantially claimed in claim 1.

Although O'Conner discloses a processor that is capable of processing more

than one instruction at a time (Col. 1, lines 16-17), as explained in the Appeal Brief, with regard to execution of one instruction by the first execution unit, O'Conner clearly discloses that one execution unit waits for results from another execution unit in order for the first execution unit to begin execution. As such, the execution units do not operate concurrently to achieve execution of an instruction.

In particular, in FIG. 1 of O'Conner, the second functional unit ("execution unit (20), as characterized by the Examiner) clearly does not operate to:

(i) generate and input data to the first functional unit ("execution unit (10)", as characterized by the Examiner) during execution of the instruction by the first functional unit (10) or

(ii) process output data received from the first functional unit (10) during execution of the instruction by the first functional unit (10), as substantially claimed in claim 1.

Thus, in view of the above, and for previous reasons set forth in Appellants' Appeal Brief, claims 1-5 are allowable over and not anticipated by O'Conner.

McNeill does not Anticipate the Claimed inventions

In the Response to Argument section of the Answer, the Examiner responds to Appellants' arguments regarding McNeill by noting that "nowhere does appellant specification or claim define the distinction between 'instruction' and 'task'" (see Answer, p. 9, section (10)). However, in the Appeal Brief, Appellants distinguish the subject matters of claims 1 and 4, for example, from McNeill by explaining that:

The rejection based on McNeill ignores the basic distinction between

> an instruction and a computing task. An instruction is a well-defined operation that takes a known number of clock cycles of a functional unit. A computing task, on the other hand, is performed by executing a series of instructions in sequence. The time required to perform the computing task is often indeterminate.
>
> In McNeill, a slave CPU performs a search on data supplied through a disk interface. Such a search is a computing task. It is not "an instruction" as set forth in the present claims.

In view of the above, the Examiner's assertion that "nowhere does appellant specification or claim define the distinction between 'instruction' and 'task'" is misplaced and way off point because the claim language in claims 1 and 4, for example, specifically recite "executing" or "execution of" an instruction.

Despite this clear language, however, the Examiner fails to properly interpret the claim language by providing a supporting explanation as to how "executing an instruction" within the scope of the claims is the same as "executing a search task" within the context of search system as taught by McNeill.

In particular, McNeill teaches in FIG. 2 a parallel searching system (20) having a master CPU processor (210) for controlling a plurality of slave CPU processors (212) for independently processing respective search requests in the search system (20) (see, e.g., Col. 3, line 65 – Col. 4, line 4). McNeill teaches that the parallel processing search system (20) is designed to perform one component of the task of database access--that of searching for the location of the requested data. The parallel nature of the architecture allows multiple database access requests to be processed simultaneously, where each slave processor (212) can search different keys and each could, if necessary, search by a different algorithm (see, e.g., Col. 9, lines 27-35).

In view of the above, one of ordinary skill in the art would readily appreciate and understand the fundamental difference between "executing instructions" on a micro-code level (as contemplated by the claimed inventions) and "performing a search task" (as contemplated in the search system of McNeill). Again, at the very least, the Examiner fails to address this basic distinction, but rather characterizes the "execution of an instruction" as being the same as "performing a search task", without providing any reasonable explanation or justification for such finding. For at least this reason, the rejection of claims 1 and 4 is legally deficient as a matter of law and fact.

The Examiner's characterization of McNeill as applied to the claimed inventions is clearly erroneous as a matter of fact for additional reasons. For instance, with regard to claims 1 and 4, the Examiner's characterization of the disk manager 16, 22 in FIG. 1 of McNeill as being the claimed "second functional unit" is not well founded. One of ordinary skill in the art would not consider the disk manager 16, 22 of McNeill as being a "second functional unit" that operates to process output data from a first function unit (slave CPU) during execution of a search task by the first function unit (slave CPU) or otherwise generate data that is input to the first function unit (slave CPU) during execution of a search task by the first function unit (slave CPU).

Indeed, there is nothing in McNeill that teaches or suggests that the disk manager 16, 22 processes data or generates data. It is well known that a disk manager (16) merely controls and manages the storage and retrieval of data to and from storage disks 12 in response to access commands. The disk manager 16 does

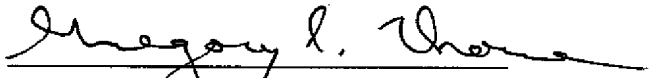
not "process data" or otherwise "generate data" that is output from or input to a functional unit for executing an instruction. As such, the Examiner's characterization of the disk manager 16 as being a "functional unit" within the scope and meaning of claims 1 and 4 for example, is improper as a matter of law and fact, and is simply a strained interpretation offered by the Examiner in an improper attempt to meet the claimed features.

Thus, in view of the above, and for previous reasons set forth in Appellants' Appeal Brief, claims 1-5 are allowable over and not anticipated by McNeill.

CONCLUSION

In view of the above, and in view of the Appeal Brief, Appellants respectfully assert that the rejection of Claims 1-5 should be reversed.

Respectively submitted,



Gregory L. Thorne
Reg. 39,398
Att'y for Applicant(s)

THORNE & HALAJIAN, LLP
Applied Technology Center
111 West Main Street
Bay Shore, NY 11706
Phone: (631) 665-5139
Fax: (631) 665-5101